



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/699,486	10/31/2003	Eric Anderson	200207252-1	3149
22879 7590 06/03/2010 HEWLETT-PACKARD COMPANY Intellectual Property Administration 3404 E. Harmony Road Mail Stop 35 FORT COLLINS, CO 80528				
EXAMINER RADTKE, MARK A				
ART UNIT 2165		PAPER NUMBER		
NOTIFICATION DATE 06/03/2010		DELIVERY MODE ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
ipa.mail@hp.com
laura.m.clark@hp.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte ERIC ANDERSON

Appeal 2009-005505
Application 10/699,486
Technology Center 2100

Decided: June 1, 2010

Before JOSEPH L. DIXON, STEPHEN C. SIU, and
DEBRA K. STEPHENS, *Administrative Patent Judges*.

SIU, *Administrative Patent Judge*.

DECISION ON APPEAL
STATEMENT OF THE CASE

This is a decision on appeal under 35 U.S.C. § 134(a) from the Examiner's rejection of claims 1, 2, and 4-24. Claim 3 is canceled.¹ We

¹ Claim 5 depends on canceled claim 3, and is thus deficient even if otherwise allowable.

have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

Invention

The invention relates to the organization of storage systems and improving storage system functionality (Spec. 2, ¶ [0001]).

Independent claim 1 is illustrative:

1. A method of creating a filesystem with transaction based functionality, comprising:

receiving an indicator to initiate a transaction for files stored in one or more portions of the filesystem;

duplicating the filesystem within a pseudo-filesystem;

creating a control text file that provides a textual filesystem interface and receives text-based commands to operate on the pseudo-filesystem;

processing the text-based commands written to the control text file; and

operating on one or more portions of the pseudo-filesystem within a transaction according to the text-based commands.

References

The Examiner relies upon the following references as evidence in support of the rejection:

Deshayes	US 6,047,294	Apr. 4, 2000
Verma	US 6,856,993 B1	Feb. 15, 2005
		(filed Mar. 30, 2000)

Brian Berliner, *CVS II: Parallelizing Software Development*, Proceedings of the USENIX Winter 1990 Technical Conference (1990) (“Berliner”).

H.T. Kung and John T. Robinson, *On Optimistic Methods for Concurrency Control*, ACM Transactions on Database Systems, Vol. 6, No. 2, pp. 213-26 (June 1981) (“Kung”).

Rejections

Claims 1, 2, 4-15, and 17-24 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Verma, Berliner, and Deshayes.

Claim 16 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Verma, Berliner, Deshayes, and Kung.

ISSUES

Issue 1

The Examiner finds that “[b]ecause Verma, exists on top of a file system, it is a pseudo-file system” (Ans. 13). The Examiner also finds that “Deshayes is directed towards backing up and restoring virtual partitions” (Ans. 16).

Appellant argues that “Deshayes does not even mention duplicate filesystems or using a pseudo-filesystem” (App. Br. 17).

Issue: Did the Examiner err in finding that Verma, Berliner, and Deshayes would have taught or suggested duplicating a filesystem within a pseudo-filesystem?

Issue 2

The Examiner finds that “Berliner teaches creating a control text file that provides a textual filesystem interface and receives text-based commands to operate on the pseudo-filesystem” (Ans. 5).

Appellant argues that “Berliner does discuss mapping locations of software code, but this teaching in Berliner has nothing whatsoever to do with duplicating a filesystem” (App. Br. 18).

Issue: Did the Examiner err in finding that Verma, Berliner, and Deshayes would have taught or suggested creating a control text file that provides a textual filesystem interface and receives text-based commands to operate on a pseudo-filesystem and operating on one or more portions of a pseudo-filesystem within a transaction according to text-based commands?

Issue 3

The Examiner finds that “[c]hanges made through the pseudo-filesystem (such as checkins of Berliner or transaction commits of Verma) result in changes to the underlying file system” (Ans. 17).

Appellant argues that “[n]owhere does Verma teach or even suggest two filesystems, namely a first filesystem and then a pseudo-filesystem wherein a file modified in the pseudo-file system is updated to the filesystem” (App. Br. 19).

Issue: Did the Examiner err in finding that Verma, Berliner, and Deshayes would have taught or suggested performing a text-based command to modify a file in a pseudo-filesystem and then updating a filesystem to include modifications performed to the file in the pseudo-filesystem?

Issue 4

The Examiner finds that Verma teaches “creating a status text file that provides text-based status results from operations performed on the pseudo-filesystem” (Ans. 7).

Appellant argues that the Examiner ignores the recitation “updates a status file [associated] with the pseudo-filesystem with a text-based status result for performing the text-based command and updates performed in the filesystem” (App. Br. 20).

Issue: Did the Examiner err in finding that Verma, Berliner, and Deshayes would have taught or suggested updating a status file associated with a pseudo-filesystem with a text-based status result for performing a text-based command and updates performed in the filesystem?

Issue 5

The Examiner finds that Deshayes teaches mounting a backed up filesystem or pseudo-filesystem (Ans. 18).

Appellant argues that “nowhere does Deshayes teach or suggest ‘mounting the entire copy of the filesystem under the pseudo-filesystem’” (App. Br. 21).

Issue: Did the Examiner err in finding that Verma, Berliner, and Deshayes would have taught or suggested creating and mounting an entire copy of a filesystem under a pseudo-filesystem?

FINDINGS OF FACT

The following Findings of Fact (FF) are shown by a preponderance of the evidence.

1. Verma teaches that “none of [the] changes made by the system for a given transaction are visible to other transactions while the modifying transaction is still active, and only become visible after the modifying transaction successfully commits” (col. 3, ll. 3-7).
2. Verma teaches that “TxF [Transactional New Technology File System] component 70 acts as the resource manager for transactional file system operations. . . . API [application programming interface] calls (e.g., CreateFileEx 80 and other file system operations) to the file system 62 generate callouts to the TxF component 70” (col. 7, ll. 10-14).
3. Verma teaches that “[d]ata changes are . . . logged by separating the operational events into one log and the actual data write details of the transaction into another log” (col. 2, ll. 57-59).
4. Deshayes teaches a “[m]ethod and apparatus for backing up and restoring data in a computer system A segment of data, such as a virtual disk partition, may be backed up at a physical level from a primary storage device to a backup storage device” (abstract).
5. Deshayes teaches “the backup storage system 54 may read information from the storage system 52 by mounting or importing virtual volumes” (col. 12, ll. 54-56).

6. Berliner teaches “the checkin program adds . . . new and changed vendor’s files to the already existing source repository The cvs ‘join’ command is a useful tool that aids [the file merging] process” (p. 5).
7. The Specification’s background of the invention section teaches that “[o]perating systems use filesystems to organize data in logical units that applications and users can easily use and manipulate” (p. 2, ¶ [0002]).
8. Appellant does not challenge the Examiner’s finding that “the use of scripts to automate certain tasks is extremely well-known in the art of Unix systems programming” (Ans. 5).

PRINCIPLES OF LAW

Claim interpretation

“In the patentability context, claims are to be given their broadest reasonable interpretations. . . . [L]imitations are not to be read into the claims from the specification.” *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993) (citations omitted). A claim meaning is reasonable if one of ordinary skill in the art would understand the claim, read in light of the specification, to encompass the meaning. *See In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004).

Obviousness

The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art,

(2) any differences between the claimed subject matter and the prior art, and (3) the level of skill in the art. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. *In re Merck & Co., Inc.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986).

“The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results,” *KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398, 416 (2007), especially if the combination would not be “uniquely challenging or difficult for one of ordinary skill in the art,” *Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007) (citing *KSR*, 550 U.S. at 418).

ANALYSIS

Issue 1

Appellant argues that Deshayes does not relate to duplicating filesystems or using a pseudo-filesystem. The Examiner’s rejection is based on the combination of Verma and Deshayes. Based on Appellant’s arguments in the Appeal Brief, we will decide the appeal of claims 1, 2, 4-11, 19, 21, and 23 with respect to issue 1 on the basis of claim 1. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Verma teaches a transactional filesystem in which changes are not visible to other transactions until the modifying transaction successfully commits (FF 1). Because these changes are seen only by the modifying transaction, the filesystem created with a transaction is a pseudo-filesystem (*id.*). Deshayes teaches backing up and restoring data in a computer system

(FF 4). Data may be backed up from sources such as virtual disk partitions (*id.*).

We are not persuaded by Appellant's argument that Deshayes does not disclose or suggest duplicating filesystems or using pseudo-filesystems. Appellant admits that a filesystem is something used by an operating system to organize data in logical units that applications and users can easily use and manipulate (FF 7). Because filesystems organize data, an artisan would have recognized that the data backup and restoration teachings are applicable to Verma's filesystem teachings. Therefore, we find that Verma, Berliner, and Deshayes would have taught or suggested duplicating a filesystem (backing up data) within a pseudo-filesystem (where changes can only be seen by a modifying transaction).

For at least these reasons, we find no evidence persuasive of error in the Examiner's 35 U.S.C. § 103(a) rejection of claim 1, and claims 2, 4-11, 19, 21, and 23 which fall therewith with respect to this issue.

Issue 2

Appellant argues that Berliner does not relate to duplicating a filesystem. The Examiner finds that Berliner teaches the creation of a control text file. Based on Appellant's arguments in the Appeal Brief, we will decide the appeal of claims 1, 2, 4-11, 19, 21, and 23 with respect to issue 2 on the basis of claim 1. *See* 37 C.F.R. § 41.37(c)(1)(vii).

As discussed above, Verma and Deshayes would have taught or suggested duplicating a filesystem within a pseudo-filesystem. Verma further teaches transactional filesystem operations (FF 2). These operations

are accessed through application programming interface commands and operated on through the use of callouts to a filesystem component (*id.*).

Berliner's teachings relate to text-based commands, such as the checkin program and the cvs join command (FF 6). These commands relate to managing source repositories (i.e., data stores) (*id.*). Appellant does not challenge the Examiner's finding that using scripts to automate tasks is well-known (FF 8). Therefore, we find that Verma, Berliner, and Deshayes would have taught or suggested creating a control text file (a script) that provides a textual filesystem interface and receives text-based commands (identifying programs or commands) to operate on a pseudo-filesystem (for transactional filesystem operations) and operating on (accessing a component) one or more portions of a pseudo-filesystem within a transaction (where changes are only visible within the modifying transaction) according to text-based commands (provided by the script).

For at least these reasons, we find no evidence persuasive of error in the Examiner's 35 U.S.C. § 103(a) rejection of claim 1, and claims 2, 4-11, 19, 21, and 23 which fall therewith with respect to this issue.

Issue 3

Appellant argues that Verma fails to teach or suggest two filesystems. Based on Appellant's arguments in the Appeal Brief, we will decide the appeal of claims 12-18, 20, 22, and 24 with respect to issue 3 on the basis of claim 12. *See* 37 C.F.R. § 41.37(c)(1)(vii).

As discussed above, Verma, Berliner, and Deshayes would have taught or suggested operating on one or more portions of a pseudo-

filesystem within a transaction according to text-based commands. Furthermore, Verma teaches that changes made by the system for a given transaction are only visible to other transactions after the modifying transaction successfully commits (FF 1). Therefore, Verma, Berliner, and Deshayes would have taught or suggested performing a text-based command to modify a file in a pseudo-filesystem (operating on one or more portions of a pseudo-filesystem according to text-based commands) and then updating a filesystem to include modifications performed to the file in the pseudo-filesystem (committing the modifications).

For at least these reasons, we find no evidence persuasive of error in the Examiner's 35 U.S.C. § 103(a) rejection of claim 12, and claims 13-18, 20, 22, and 24 which fall therewith with respect to this issue.

Issue 4

Appellant argues that the Examiner does not address updating a status file associated with a pseudo-filesystem. The Examiner initially rejected claim 12 by reference to the "remarks and discussions made in claim 1" (Fin. Rej. 7). However, the Examiner now also makes reference to claim 8 (Ans. 8). Based on Appellant's arguments in the Appeal Brief, we will decide the appeal of claims 12-18, 20, 22, and 24 with respect to issue 3 on the basis of claim 12. *See* 37 C.F.R. § 41.37(c)(1)(vii).

As discussed above, Verma, Berliner, and Deshayes would have taught or suggested operating on one or more portions of a pseudo-filesystem within a transaction according to text-based commands. Verma also teaches logging of data changes as operational events and actual data

write details (FF 3). Therefore, Verma, Berliner, and Deshayes would have taught or suggested updating a status file (logging) associated with a pseudo-filesystem with a text-based status result (actual data write details) for performing a text-based command (operating on a pseudo-filesystem according to text-based commands) and updates performed (operational details) in the filesystem.

For at least these reasons, we find no evidence persuasive of error in the Examiner's 35 U.S.C. § 103(a) rejection of claim 12, and claims 13-18, 20, 22, and 24 which fall therewith with respect to this issue.

Issue 5

Appellant argues that Deshayes does not teach mounting an entire copy of a filesystem under a pseudo-filesystem. The Examiner holds that Deshayes teaches mounting a backed up filesystem or pseudo-filesystem.

Deshayes teaches backing up and restoring data, such as data on a virtual disk partition (FF 4). Deshayes also teaches that information from the storage system may be read by mounting or importing virtual volumes (FF 5). As discussed above, Verma teaches pseudo-file systems (FF 1). Therefore, we find that Verma, Berliner, and Deshayes would have taught or suggested creating (backing up) and mounting (enabling reading of) an entire copy of a filesystem (a virtual disk partition) under a pseudo-filesystem (where changes are not visible to other transactions).

For at least these reasons, we find no evidence persuasive of error in the Examiner's 35 U.S.C. § 103(a) rejection of claim 13 with respect to this issue.

CONCLUSIONS OF LAW

Based on the findings of facts and analysis above, we find no evidence persuasive of error in the Examiner's findings:

1. that Verma, Berliner, and Deshayes would have taught or suggested duplicating a filesystem within a pseudo-filesystem (issue 1);
2. that Verma, Berliner, and Deshayes would have taught or suggested creating a control text file that provides a textual filesystem interface and receives text-based commands to operate on a pseudo-filesystem and operating on one or more portions of a pseudo-filesystem within a transaction according to text-based commands (issue 2);
3. that Verma, Berliner, and Deshayes would have taught or suggested performing a text-based command to modify a file in a pseudo-filesystem and then updating a filesystem to include modifications performed to the file in the pseudo-filesystem (issue 3);
4. that Verma, Berliner, and Deshayes would have taught or suggested updating a status file associated with a pseudo-filesystem with a text-based status result for performing a text-based command and updates performed in the filesystem (issue 4); and
5. that Verma, Berliner, and Deshayes would have taught or suggested creating and mounting an entire copy of a filesystem under a pseudo-filesystem (issue 5).

DECISION

We affirm the Examiner's decision rejecting claims 1, 2, and 4-24 under 35 U.S.C. § 103(a).

Appeal 2009-005505
Application 10/699,486

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

msc

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS CO 80528